

Worcester Polytechnic Institute

Digital WPI

Interactive Qualifying Projects (All Years)

Interactive Qualifying Projects

2019-09-05

Dynamic Imagery

Keith Thabo Blackstock
Worcester Polytechnic Institute

Follow this and additional works at: <https://digitalcommons.wpi.edu/iqp-all>

Repository Citation

Blackstock, K. T. (2019). *Dynamic Imagery*. Retrieved from <https://digitalcommons.wpi.edu/iqp-all/5551>

This Unrestricted is brought to you for free and open access by the Interactive Qualifying Projects at Digital WPI. It has been accepted for inclusion in Interactive Qualifying Projects (All Years) by an authorized administrator of Digital WPI. For more information, please contact digitalwpi@wpi.edu.

Dynamic Imagery

And the Narrative Power of Pictures

An Interactive Qualifying Project Report
Submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
In cooperation with Team OCULUS SDA 2019

Project Advisor: Tahar El-Korchi, Ph.D, Mohammed Brahimi, Ph.D
By: Keith T. Blackstock
August 2019

ABSTRACT

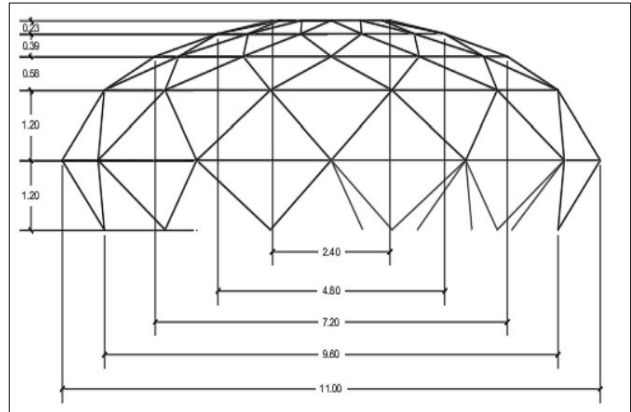
The goal of this project was to create an active outreaching social media platform to meet the requirements of the Solar Decathlon Africa (SDA) competition in addition to informing the general public about Team Oculus' progress.

TEAM OCULUS

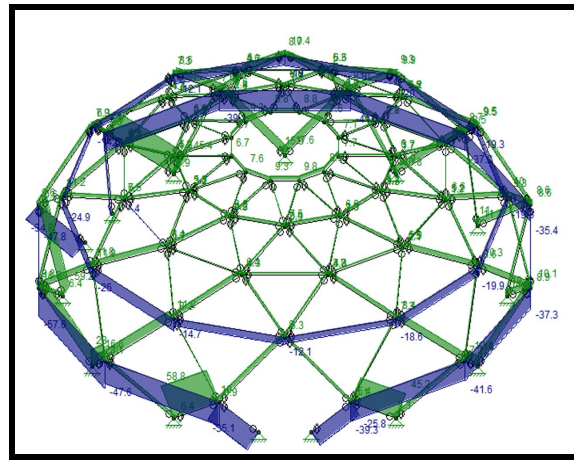
Comprising of students of WPI, ENSAM (Meknes), ENSAIS (Rabat), and Abuja (Nigeria), this intercollegiate team has collaborated toward the ideation, schematic development, and functional modeling of a net-zero solar energy home taking the form of an augmented geodesic dome. Per namesake, the Solar Decathlon competition assesses each entry on ten design criterion, ranging from numerical metrics (electrical performance, operational efficiency, systems integration), commercial prospects (building costs, market potential, comfort, weather resilience, technical innovation), and aesthetic appeal (presentation, architectural beauty). To see that all the design challenges were met, students from all three universities split into various groups, each tackling a specific objective — all working in concert toward the final product.

THE GEODOME

The idea surrounding the overall design of the housing structure played on both the structural efficiency of a geodesic dome, and the elegant simplicity of the traditional African hut. The design also features an adjusted aspect ratio (lower ceiling, wider layout), offering certain practical and aesthetic advantages over a simple hemispheric design.



1



2

Computational models of the design — like the one above, generated in ASCE 7-10 — confirmed the structural viability of the team's particular design; five ascending truss-rings, two pentagonal openings, and a central “oculus” for ambient lighting.

This “eye” is the design that Team Oculus envisions — pardon the pun. The remaining design features were developed in parallel; the utilities and living-quarters arrangement, electrical wiring, HVAC system, plumbing scheme, external weatherproof envelope — and of course, the photovoltaics, mounted on top of an external garage housing an electric car.

¹ Figure 1: 2-D Revit truss profile — rendered by mechanical engineering subgroup.

² Figure 2: ANSYS ASCE 7-10 3-D beam loading scheme — rendered by mech. engineering subgroup.

DEFINING *DYNAMIC IMAGERY*

For our purposes, a dynamic image is a picture within a picture — really, any additional content embedded within a picture, that is displayed on some sort of command. In our case, cursor position serves as said command.



So if, for instance, one were to hover their cursor over a particular file laid out on the desk depicted in the image above, an expanded version of it would appear behind Professor Van Dessel; in addition, perhaps, to some written or recorded dialogue further explaining what is being depicted.

There were three main reasons behind taking this direction with the project. First off, it's cool — which is actually quite important where public outreach is concerned. Secondly, the audience can view more content within a given photo. Thirdly, a *visual narrative* is created, calling for a greater degree of audience engagement, which leads us into a discussion about interactive learning.

INTERACTIVE LEARNING

There is a wealth of scholarly data supporting the merits of interactive learning — and I’m not strictly talking in an academic sense. Consider a museum. A well designed exhibit — that is, one that strongly encourages visitors to explore its content — plays on the principle of interactivity. It’s not uncommon to see buttons on one or both sides of a plaque beside a glass display, that, when pressed, illuminates a certain area in that display, along with some recorded curation.

Anyone could read through a set of texts and gain a fairly detailed understanding of pretty much anything; but the appeal of the museum is the exploratory manner in which the information is sought and received.

As Cuberis author, Eric Holter put it, “visual interface elements that manifest the oblique connections between artworks, artists, events, posts, audio resources, research, and other related content are not only helpful user-experience tools, they reveal the depth and richness of museum content.”, (The Visual Display of Museum Impact). It simply isn’t enough to “put stuff out there” for people to stumble upon. If the goal is to pique the audience’s curiosity, you must create an *experience* within your content, and interactivity brings this element.

METHODS

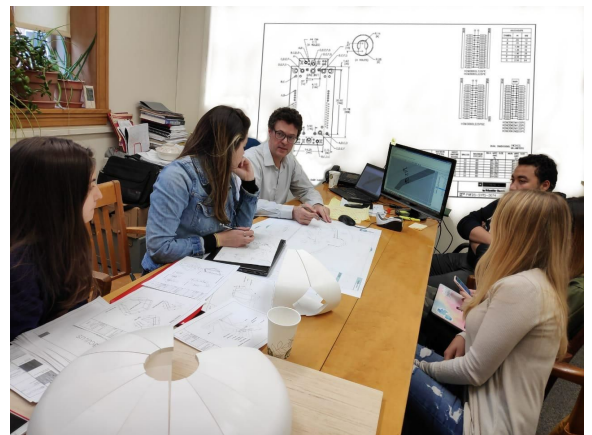
The creation of these dynamic images is twofold — one part graphic editing, and one part Javascript programming. Before any of that, however, I need a keyframe. The keyframe is a static image, selected from one of Team Oculus’ myriad photo albums, with subject matter being paramount. Even with all the extra content I’ll be embedding, the keyframe needs to meet certain criteria. In other words, I need something to work with.

- ❖ Workable resolution. Obviously...
- ❖ Two or more team members depicted; preferably facing either each other, or some common item(s).

- ❖ Action. Candid shots taken mid-conversation are golden, as are those depicting the assembly of the house.
- ❖ Space. While activity is important, a hyperactive, “busy” frame doesn’t leave much room for the *dynamic* aspects that come later.

Once I have selected a keyframe, I import it into a graphic editor, my weapon of choice was Autodesk Sketchbook. The specific operations done on the images inside of Sketchbook vary with the desired dynamic effect. Invariably, there’s a considerable amount of cropping, duplicating, and merging in each case.

Let’s return to the earlier image, or rather, the four images used to create the dynamic effect.



As the cursor passes over the four central schematic documents, the Javascript program reads its local pixel coordinates and switches to the corresponding image. Because everything but the backdrop remains the same, the effect is that of a seamless transition.

Here's a snippet of the source code.

```
console.log("hi Keith: " + pageX, pageY);

if (((pageX > 622) && (pageX < 812)) && ((pageY > 450) && (pageY < 620))) {

    console.log("target1");
    changeBGImage("drawing_board.J3.jpg", ".mp3" );

} else if (((pageX > 300) && (pageX < 500)) && ((pageY > 550) && (pageY < 650))) {

    console.log("target2");
    changeBGImage("drawing_board.J.jpg", ".mp3" );

} else if (((pageX > 510) && (pageX < 600)) && ((pageY > 500) && (pageY < 570))) {

    console.log("target3");
    changeBGImage("drawing_board.J2.jpg", "" );

} else {

    changeBGImage("drawing_board.J0.jpg", "" );

}
```

Note the “.mp3” file fields. The event handler³ structure is equipped to que audio clips as well as visual data. In any given scene, there's rich information beyond that which the viewer can immediately see; a dynamic image allows the depicted team members to deliver some of that information — to curate the scene.

In addition to the Javascript, a bit of CSS⁴ (Cascading Style Sheets) code handles the navigation bar used to transition between the various scenes.

³ a software routine that processes actions such as keystrokes and mouse movements.

⁴ a stylesheet language used to describe the presentation of a document written in HTML or XML.

The Oculus website is separated into several scenes. Each scene is a separate HTML file. These scenes all use a single navigation component. The navigation component allows the user to move backwards and forwards through out the site. The navigation component was written in a way allowing Oculus site developers an easy way to add addition scenes (HTML files), by modifying a single navigation component.

Project Oculus	Draft Phase	Structural Assembly	Bracing and Fitting	scene 4	More Information.
----------------	-------------	---------------------	---------------------	---------	-------------------

Each scene (HTML file) references the navigation component as follows:

```
<html>

<head>

<title>Oculus Project</title>

<meta charset="utf-8"/>

<link rel="stylesheet" type="text/css" href="navbar.css" />

</head>

....

<body>

<div id="navbar"></div>

<script src="navbar.js"></script>

....
```

The navigation bar is rendered inside the scene HTML file via the **navbar.js** JavaScript file. If a new scene is created the Oculus site developer should add the scene title and html file in the bolded productList and linkList arrays shown below. There is a one to one correspondence between the two arrays associating the navbar text with its HTML page.

```
(function () {

var ul = document.createElement('ul');

ul.setAttribute('id', 'proList');
```



```

productList = ['Project Oculus', 'Draft Phase', 'Structural Assembly',
'Bracing and Fitting', 'scene 4', 'More Information.'];

linkList = ['home.html', 'scene1.html', 'scene2.html', 'scene3.html',
'scene4.html', 'more.html'];

document.getElementById('navbar').appendChild(ul);

productList.forEach(renderProductList);

function renderProductList(element, index, arr) {

    var li = document.createElement('li');

    li.setAttribute('class', 'item');

    var a = document.createElement('a');

    var linkText = document.createTextNode(element);

    a.appendChild(linkText);

    // a.title = "my title text";

    a.href = linkList[index];

    // a.href = "http://example.com";

    li.appendChild(a);

    ul.appendChild(li);

    // li.innerHTML=li.innerHTML + element;

}

})();

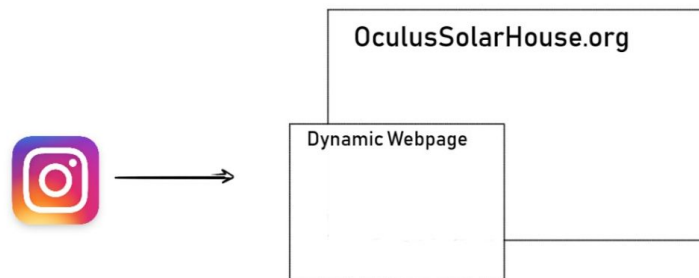
```

SELECTED KEYFRAMES



IMPLEMENTATION

All of this web-based content exists, as of this moment, in a private server I licensed through Heroku, a cloud platform created for open-source programmers. Upon full completion, the project will be transferred over to the official Team Oculus webpage, where it can exist alongside the rest of the team's public content. This will streamline the social media linking process. One could post a still image to, say, Instagram, and embed a link inside of said post, directing the viewer to the dynamic web-program, operating within the main Team Oculus site.



CONCLUSION

It is beyond the scope of this write-up to document how to write HTML, CSS, and JavaScript. However, the documentation above should instruct an Oculus developer on how to make changes to the site with basic website development knowledge. As previously mentioned (SH. *INTERACTIVE LEARNING*), interactive content is widely agreed as a viable and effective means of public outreach. Furthermore, I have published the code involved in creating these *dynamic images*, and all necessary technical components will be available to this team, as well as future WPI Solar Decathlon teams. In other words, the entire framework is in place; the rest is a matter of content.

REFERENCES

Pınar Nuhoğlu Kibar, Buket Akkoyunlu. (2017) Fostering and assessing infographic design for learning: the development of infographic design criteria. *Journal of Visual Literacy* 36:1, pages 20-40.

Terry Loerts, Christina Belcher. (2019) Developing visual literacy competencies while learning course content through visual journaling: teacher candidate perspectives. *Journal of Visual Literacy* 38:1-2, pages 46-65.

Holter, E. (2019, May 16). The Visual Display of Museum Impact - Cuberis - Museum Website Design and Development. Retrieved from <https://cuberis.com/the-visual-display-of-museum-impact/>

To view the existing content, visit <https://wpioculus.herokuapp.com>.

All of the source code, images, and audio clips can be accessed in this [GitHub file repository](#).